NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER

DEFENSE COMMUNICATIONS AGENCY

COMPUTER SYSTEMS MANUAL
CSM AM 9A-67
VOLUME I
21 FEBRUARY 1972

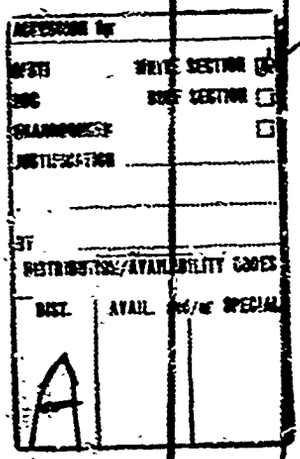
THE NMCSSC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK)

DATA INPUT SUBSYSTEM

ANALYTICAL MANUAL

RECORD OF CHANGES

CHANGE NUMBER	DATED	DATE ENTERED	SIGNATURE OF PERSON MAKING CHANGE



UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
National Military Command System Support Center (NMCSSC)
Defense Communications Agency (DCA)
The Pentagon
Washington, DC 20301

2a. REPORT SECURITY CLASSIFICATION

2b. GROUP

3. REPORT TITLE
The NMCSSC Quick-Reacting General War Gaming System (QUICK)
Analytical Manual, Volume I Data Input Subsystem

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
N/A

5. AUTHOR(S) *(First name, middle initial, last name)*
NMCSSC: Peter K. Cook
Dianne Hanak
Donald F. Webb
Lambda Corp: Betty J. Ellis
Jack A. Sasseen

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO OF REFS
21 February 1972	47	4

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
DCA 100-70-C-0065	NMCSSC
b. PROJECT NO. NMCSSC Project 631	COMPUTER SYSTEM MANUAL CSM AM 9A-67
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT
This document is approved for public release; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	National Military Command System Support Center/Defense Communications Agency The Pentagon, Washington, DC 20301

13. ABSTRACT

This is one of three volumes describing the analytical aspects of the Quick-Reacting General War Gaming System (QUICK). This volume addresses the system data requirements, the organization and structure of the data base, and the concepts and techniques employed to prepare a game data base for subsequent Plan Generation and Simulation. In addition, it identifies the major system limitations and discusses accuracy considerations.

Based upon a suitable data base and user control parameters, QUICK will generate individual bomber and missile plans suitable for war gaming, and simulate the planned events. The generated plans are of a form suitable for independent review and revision. Subsequently, the planned events are simulated; various statistical summaries are produced to reflect the results of the war game. A variety of force postures and strategies can be accommodated.

QUICK is documented extensively in a set of Computer System Manuals (series 9-67) published by the National Military Command System Support Center (NMCSSC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

UNCLASSIFIED

NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER

Computer System Manual Number CSM AM 9A-67

21 February 1972

THE NMCSSC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK)

Analytical Manual

Volume I - Data Input Subsystem

Submitted by:

Donald F. Webb

DONALD F. WEBB
Major, USAF
Project Officer

REVIEWED BY:

R. E. Harshbarger

R. E. HARSHBARGER
Technical Director
NMCSSC

APPROVED BY:

Bruce Merritt

BRUCE MERRITT
Colonel, USA
Commander, NMCSSC

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.

This document is approved for public release; distribution unlimited.

ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Development and Analysis, NMCSSC, in response to a requirement of the Studies, Analysis and Gaming Agency (SAGA), Organization of the Joint Chiefs of Staff. Technical support was provided by Lambda Corporation under Contract Number DCA 100-70-C-0065.

CONTENTS

ILLUSTRATIONS

TABLES

ABSTRACT

This is the first of three Computer System Manuals describing the analytical aspects of the Quick-Reacting General War Gaming System (QUICK). It addresses the system data requirements, the organization and structure of the data base, and the concepts and techniques employed to prepare a game data base to support a specific plan generation/simulation requirement. In addition, applicable constraints and accuracy considerations are described.

QUICK is a two-sided strategic nuclear exchange war gaming system. It is designed to assist the military planner in examining various facets of strategic nuclear war involving a variety of forces, strategies and starting conditions. Since the volume of data required to support such investigations is substantial, the Data Input subsystem is designed to provide an efficient means of assembling, maintaining, and organizing an input data base to support user requirements for plan generation and simulation.

The Analytical Manual consists of three volumes. This volume of the Analytical Manual describes the Data Input subsystem of QUICK. Volume II describes the Plan Generation subsystem, and volume III describes the Simulation and Data Output subsystems.

The following is a list of associated documents on the QUICK system.

GENERAL DESCRIPTION
Computer System Manual CSM GD 9A-67
A nontechnical description for senior management personnel

PROGRAMMING SPECIFICATIONS MANUAL
Computer System Manual CSM PSM 9A-67 (three volumes)
Detailed information required for system maintenance and modification

USER'S MANUAL
Computer System Manual CSM UM 9-67 (two volumes)
Detailed instructions for applications of the system

OPERATOR'S MANUAL
Computer System Manual CSM OM 9A-67
Instructions and procedures for the computer operators

CHAPTER 1
INTRODUCTION

GENERAL

The QUICK-Reacting General War Gaming System (QUICK) is designed to assist in the study of strategic conflicts involving a large-scale exchange of nuclear weapons. The system is structured into four major subsystems: the Data Input, Plan Generation, Simulation, and Data Output subsystems. This first volume of the Analytical Manual describes the Data Input subsystem which performs the function of accepting required input data and assembling a game data base which is suitable for input to the Plan Generation, Simulation, and Data Output subsystems.

Data Assembly

The total volume of data required to support a variety of QUICK applications is substantial; however, the majority of these data can be preassembled and maintained in a readily accessible form. This technique is employed by NMCSSC in providing QUICK support. Using the NMCSSC QUICK Data Base Generation System (QDBGS)*, data are retrieved from various automated data source files maintained by NMCSSC and merged with other manually prepared data to create a source file (called DATADB) for the QUICK system.

This data file may be viewed as a master data base or data library, in that it contains more information than is required to produce a single set of Red and Blue plans. There are no restrictions on the quantity of data that may be maintained in this data file; however, there are constraints (upper limits) on the amount of data the QUICK system can process. While none of these upper limits is considered a significant restriction, they are addressed in greater detail in chapter 2.

Contents of Data Base (DATADB)

To provide a flexible data source, force structures corresponding to established procurement schedules and to selected intelligence projections may be maintained for both Red and Blue forces. Specifically, the data

*The QDBGS is not a component of QUICK and its operation is not described in this manual.

base contains the information required to define: (1) the capabilities and characteristics of the offensive and defensive weapon systems; (2) the physical characteristics of the installations to be considered as potential targets; (3) related geographic-type data required by the QUICK system for plan generation and simulation, e.g., data describing bomber air defense zones; and, (4) planning parameters such as the estimated probability of destruction before launch (DBL) established for each offensive weapon system (a more specific explanation of the data content and structure of the data base is presented in chapter 2).

Selection of Game Data

The NMCSSC QUICK data base, as described above, contains more data than is required for any single support task. Consequently, the Data Input subsystem is designed to provide the facilities required to abstract from the data base that information the user desires for a particular simulation or plan development task. In addition, this subsystem provides the capability to add, delete, and/or modify the selected data as required to support the specific game scenario involved. Following selection and modification, a game data base is automatically structured to meet the requirements of the Plan Generation, Simulation, and Data Output subsystems.

CONCEPT OF OPERATION

QUICK System

The following describes the general concept of operation for the QUICK system and establishes the relationship of the Data Input subsystem to the other major subsystems.

Figure 1 illustrates the procedure and information flow between the QUICK subsystems. The processing sequence is shown by solid lines and the information flow by dashed lines. Magnetic computer tapes are used to pass information between the four subsystems.

Processing is initiated by inputting the parameters which identify the Red and Blue forces and the potential targets which are to be extracted from the NMCSSC QUICK data base. In addition, any desired data base modifications are specified. The Data Input subsystem then processes the QUICK data base and prepares a simulation data tape for input to the Simulation subsystem and a game data base tape for input to the Plan Generation and Data Output subsystems.*

---

* The QUICK subsystems are also referenced by the names Input subsystem, Plan Generator, Simulator, and Output subsystem.

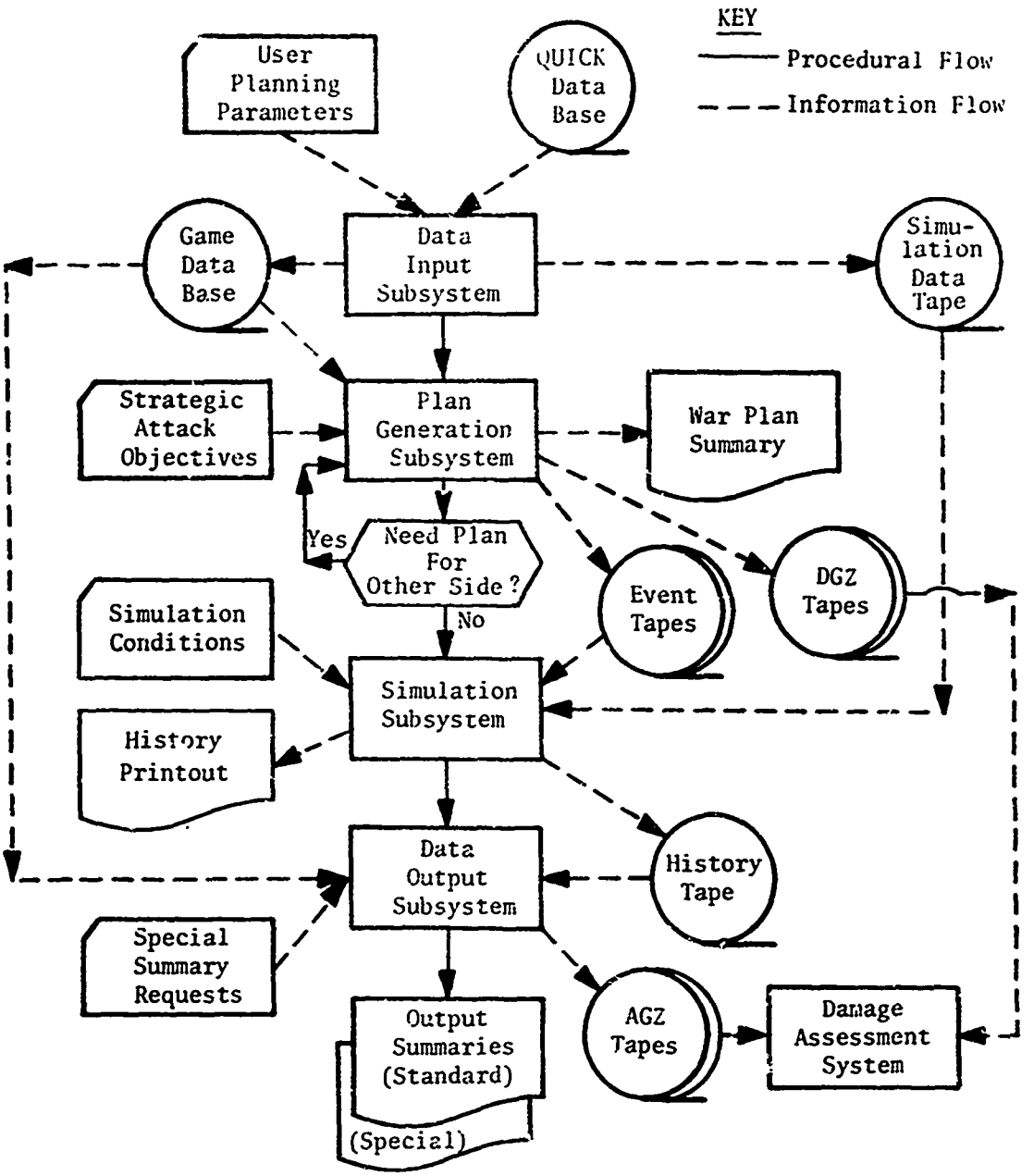


Fig. 1. Procedure and Information Flow in QUICK

The Simulator produces a History tape for input to the Data Output subsystem. The Data Output subsystem uses the History and game data base tapes to produce two special history tapes which are used within the Data Output subsystem to prepare printed summaries of the simulated events. In addition, the Data Output subsystem prepares the actual ground zero (AGZ) tapes which contain the weapon delivery data, e.g., target location and weapon yield. These tapes are subsequently processed by a damage assessment system separate from QUICK.

While the system can proceed automatically through all steps if desired, it may be halted at the end of each subsystem and the available output inspected for correctness and adequacy.

Data Input Subsystem

The Data Input subsystem is a user-oriented data management tool. The subsystem does not involve analytical techniques per se but performs data processing functions associated with updating and creating data files used in QUICK.

The Input subsystem includes the programs, data maintenance routine, and special procedures required to abstract from the QUICK data base (DATADB) the information a user desires for a particular plan or game, and assembles it in a form suitable for input to the Plan Generation, Simulation, and Output subsystems. The Input subsystem has been designed as a general-purpose, flexible and expandable system which can be modified to delete and/or add capabilities unique to a specific support task.

The major programs of the Input subsystem are shown in figure 2. A brief description of the flow of data and principal functions of the programs involved will provide a basis for understanding the more detailed descriptions which appear later in this volume.

Program QUIKBASE: This program performs the primary function of creating a game data base which defines the general data to be used in the succeeding programs of the Input subsystem. This program accepts an input data base and, based on user-input parameters, modifies the file to create a game base file (QUIKDB). The input data base (also called a data library) may be data file DATADB, created using the NMCSSC QUICK Data Base Generation System (QDBGS) or, as an alternate capability, the data base may be input in card form. On option, the game base file (QUIKDB) can be printed.

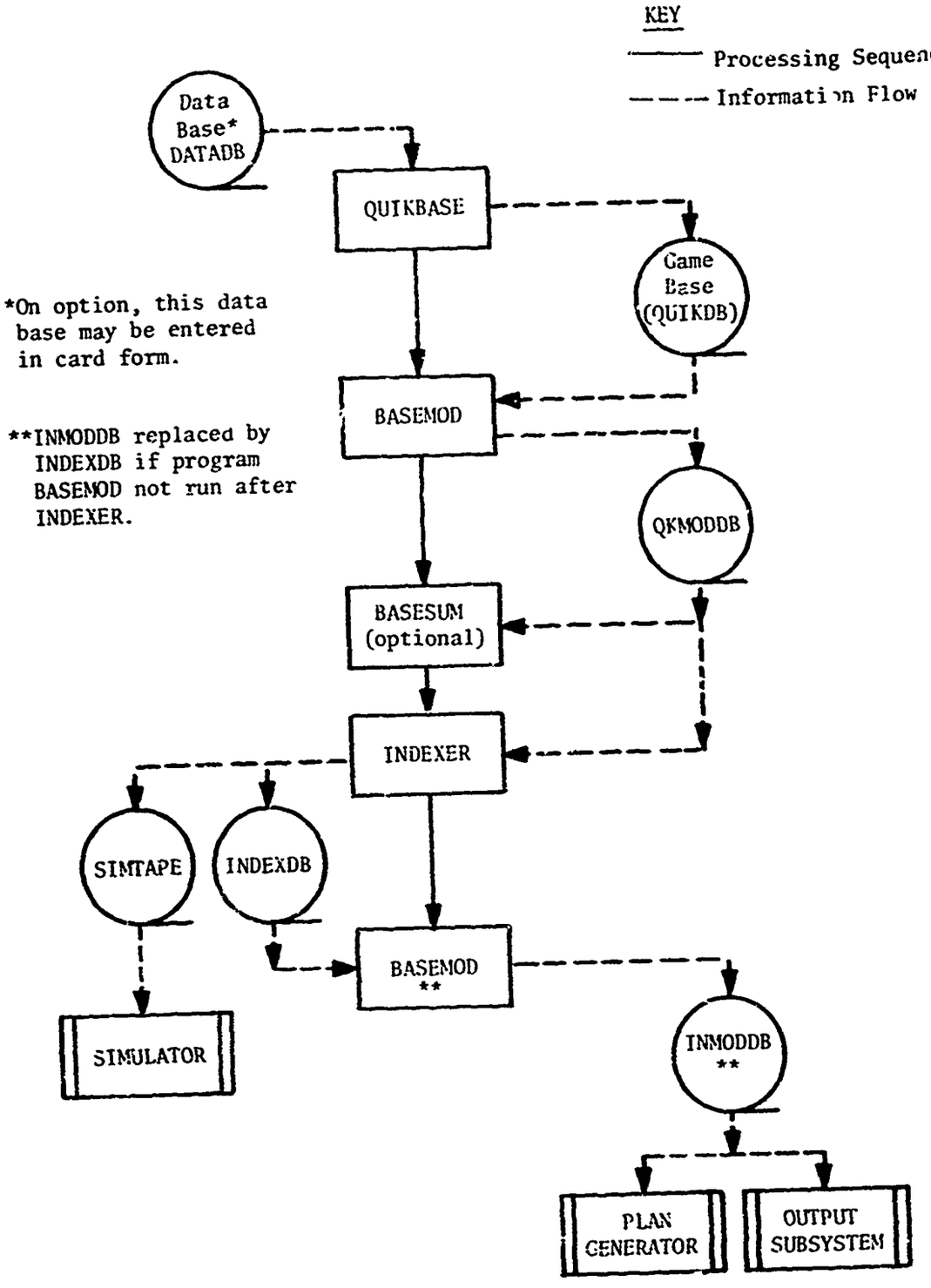


Fig. 2. Data Input Subsystem

Program BASEMOD: This program is designed to perform a specific NMCSSC support task. Its main function is to alter the content or characteristics of a data base in order to adapt it to the specific scenario for which a plan is being developed. As indicated in figure 2, this program is utilized after program QUIKBASE and/or after program INDEXER to introduce user-desired modifications in the game file. Examples of the types of modifications involved are presented in chapter 3.

Program BASESUM: The purpose of program BASESUM is to summarize a game base file and to print these summaries in tabular form. While figure 2 reflects the program operating after program BASEMOD, the program may be used to summarize the data base contained on the output tapes produced by programs QUIKBASE, BASEMOD, or INDEXER. This program, while not required in the processing cycle, provides a means for checking the input data and is a record of the information contained on any game base tape.

Program INDEXER: To provide for efficient data handling and communications between the programs of the QUICK system, index numbers are assigned to various kinds of data. For example, each item assigned to a target class in the data base (15 target classes are used in QUICK) is assigned an index number, called INDEXNO, which may range from 1 to 12,000. For internal coding purposes, such items are referenced by their assigned INDEXNO. Program INDEXER is designed to perform the required indexing operations. In addition, target elements which are either exactly collocated or are within close proximity of each other are grouped together to facilitate calculations of target kill probabilities during simulation and to simplify the weapon allocation procedures used in plan generation. These program functions are discussed in greater detail in chapter 3.

The primary NMCSSC input to program INDEXER is the modified data base prepared by program BASEMOD (QKMODDB). Program INDEXER prepares two output tapes. The simulation data tape SIMTAPE contains selected weapon and target data and is subsequently input to the Simulation subsystem. In addition, an indexed data base INDEXDB is prepared. The INDEXDB tape serves as input to: (1) the Plan Generation subsystem and (2) the Data Output subsystem. If additional modifications are required, INDEXDB is processed by program BASEMOD and a modified indexed game data base INMODDB is output for use in place of INDEXDB.

The game data base tape input to the Plan Generation subsystem contains a complete indexed list of all Red and Blue game objects; however, a single run of the Plan Generator produces a plan for only one side. Consequently, the Plan Generator must be cycled twice to produce the Red and Blue plans. Two major inputs are required to initiate this phase of processing: (1) the game data base prepared by the Data Input subsystem;

and (2) a set of parameters which relate to the strategy associated with the plan which is to be developed. These parameters are supplied by the planner. They reflect his views as to the strategic attack objectives, in terms of the relative values of the various targets being considered, the forces to be withheld, the targeting constraints to be observed, and the initiating force, i.e., which side attacks first.

The target values which are computed on the basis of these parameters reflect in a very significant way the major strategic objectives of the war plan which is to be developed. These values are relative values and are partially established in the data base itself. In the data base, each potential target is assigned a value calculated to reflect its relative worth within its assigned class (15 target classes are considered). To generate a specific plan, the user's input parameters provide data which determine the relative value of the target classes, and hence of all targets, for the current plan. The user thereby puts more or less relative importance on each of the classes of targets in accomplishing the strategic objectives that he chooses. This of course will be related to the kind of strategy he is contemplating for the particular war game, whether a first or second strike, and so forth.

Having established a value for each target, the Plan Generator then allocates the weapons, e.g., Red weapons to Blue targets, and prepares the detailed missile and bomber attack plans. The magnetic tape which reflects the series of Red missile and bomber events corresponding to the sortie plans is prepared in a form suitable for input to the Simulator. As a user option, a war plan summary is provided by the Plan Generator which includes an expected-value estimate of the results of the attack. In addition, a computer tape containing the desired ground zero (DGZ) for each planned weapon can be output to facilitate detailed damage analysis using an external damage assessment system. The war plan for the opposing side is then prepared in the same manner. The system is ready to proceed with the simulation.

The simulation conditions, specifying the starting time for each side and various defense capabilities, are read in from cards. The events on the event tapes are then processed in the Simulator. For each event that transpires, a record is made on the History tape of all information that might later prove of interest.

QUICK General-Purpose Utility Package

In addition to the main programs of the four QUICK subsystems, QUICK employs a general-purpose utility package. This utility package consists of programs, subroutines, and functions which perform a variety of support

tasks common to two or more system programs. These programs and routines are discussed in chapters 2-4 of the Programming Specifications Manual, Volume I, and in chapter 1 of the User's Manual, Volume II. Program DECLARES, a component of this utility package, performs a unique task which warrants additional comment.

Program DECLARES (Computer Program Processor): Program DECLARES is designed to aid the NMCSSC analyst in maintaining four QUICK programs which process QUICK data base tapes. The programs involved are BASEMOD and INDEXER of the Data Input subsystem, PLANSET of the Plan Generation subsystem, and Program READSUM of the Data Output subsystem.

The programming techniques and FORTRAN coding used within each of these programs are directly related to the structure and content of the directory associated with the QUICK data base (a technical explanation is included in the PSM). Changes in the directory which involve the addition, deletion, or change in the ordering of the attributes listed therein, will always require a programming modification to BASEMOD, INDEXER, PLANSET, and READSUM.

Program DECLARES provides the NMCSSC analyst with a relatively simple method of implementing program modification required as a result of changes in the directory. To use this capability a set of DECLARES command cards are inserted in the program source code (FORTRAN) deck. Program DECLARES is then used to process the program deck, effect the required modifications, and prepare an output tape containing the modified FORTRAN program. The modified program is subsequently compiled using standard procedures and is ready for execution.

Programs BASEMOD, INDEXER, PLANSET, and READSUM must be processed by program DECLARES prior to each compilation. In addition, they must be reprocessed when: (1) the order of the attributes in the directory is modified; or (2) the directory is changed by adding or deleting attributes.

CHAPTER 2
DISCUSSION OF METHODS

This chapter presents the general concepts for storing and retrieving data in the QUICK system. "Storage" here refers not only to the input data base but also the game outputs in a form convenient for output processing.

The storage and retrieval techniques used in QUICK permit the storage of data in an unstructured* file, having the properties of simplicity; freedom to add, delete, or change the form of data without being restricted by rigid file format; ability to extract any subset of the data and organize it in any specified way; ability to retrieve, organize, and summarize the data with the full generality of set theory; and reasonably efficient operation.

DATA BASE CONCEPTS

The data base is a collection of elements (items); each item is described by a number of attributes (see Appendix B, QUICK Attribute Names and Descriptions). The items may occur in any order in the data file. Sets of the items in the data base can be defined as functions of the attributes of the elements. Not every attribute is defined for every item in the data base, and only a small fraction of the total attributes occurring in the data base will be defined for any single item. For example, the attribute "CEP" will be defined only for those elements contained in the data base which represent weapon delivery systems.

It is necessary to distinguish between the name of an attribute and the value associated with that attribute. Thus, "CEP" is the name of an attribute, while the value of that attribute would be a real number such as 1.5. The values of attributes need not be numbers. For example, the attribute "CLASS" might have a value of "MISSILE" or "BOMBER."

A single item in the data base is described by a variable number of attribute pairs, the first element of each pair specifying the attribute

---

*In the sense that items in the file can occur in any order, and that all items are represented in a single general format.

name, and the second element being the value for that attribute. For example, an item representing a single Minuteman missile would be represented on the data base tape as a list of pairs (table 1) with the first element of each pair containing the index of the attribute whose value is given by the second element of the pair. (On the magnetic tape file which contains the data base, the attribute name is replaced by the index number associated with the attribute. In table 1, the associated attribute name is shown in parentheses but does not appear on the tape.) By adopting this structure wherein each attribute value is preceded by the explicit index (or position) of the attribute in an attribute name array, the order of the attribute-value pairs is immaterial. No space need be reserved for the very large number of attributes which are not defined for a particular item, as would necessarily be the case for a fixed format file structure in which the attributes were defined implicitly by position.

Table 1. Example of Minuteman Entry on the Data Base Tape

INDEX(*)	VALUE
1 (CLASS)	MISSILE
2 (TYPE)	MM-II
3 (SIDE)	BLUE
11 (NAME)	MANDAN
94 (TVUL)	.0466
28 (LAT)	42.0
29 (LONG)	101.0
85 (REL)	.79
78 (CEP)	.6
75 (PAYLOAD)	13

The logical structure of storage and retrieval operations with this file does not depend in any way upon the order of the items within the file. In addition, considerable compression of the file can be obtained by ordering the file in such a manner as to permit "global" attribute

*For attribute descriptions, see appendix B.

definitions which apply to consecutive groups of items. This ordering and compression of the file is only an efficiency device which in no way compromises the advantages in flexibility and simplicity of an unstructured file.

This form of data file is processed by a single pass of the entire file, taking appropriate action for each item as it passes through. Most operations, such as counting, extracting, or modifying, are accomplished in this manner. The processing techniques used by the programs of this subsystem are described in chapters 5-8 of the Programming Specifications Manual, Volume I.

It is to be noted that there are four kinds of attributes in the directory:

1. Attributes which are necessary if QUICK is to function accurately, e.g., LAT (latitude)

2. Attributes which are automatically generated by QUICK programs, either for internal indexing (e.g., ITYPE) or a result of simulation, e.g., DELAY

3. Attributes included for the convenience of the user, usually for identification or classification purposes, e.g., NAME and FUNCTION

4. Attributes added for possible future use and not referred to by any QUICK programs, e.g., WACNO .

DATA BASE ORGANIZATION

The QUICK data base consists of two parts, the directory and the Item File. These are described in the following paragraphs.

The Directory

The heart of the Data Input (and Output) subsystem is the directory, which consists of a list of all the attributes which can be used to describe the items defined in the data base (see Appendix B, QUICK Attribute Names and Descriptions). In addition to the mnemonic name assigned to the attribute, the directory also includes information about the range of values which may be associated with it and the value which is given to it when it is not defined for any item.

This directory facilitates the input of items to the data base, directs various kinds of checking of the attribute values included for each item, and simplifies retrieval and output programs. All items which are input to the data base are checked for consistency. That is, the attributes which are specified to describe an item are checked to determine that they are, in fact, defined in the data base directory and that the attribute value satisfies the range or checklist specifications for those attributes. In addition, the position of attributes in the directory is used as the appropriate index number assigned to the attribute. The index number is stored on the data base file instead of the attribute name to simplify processing. Instructions on the preparation of the directory are contained in chapter 2 of the User's Manual, Volume II (Program QUIKBASE, SETID Option, Directory Card Images).

Directory Conventions

In preparing the directory, certain conventions have been adopted for the units in which values of the attributes are expressed. The ones of greatest interest are: distances expressed in nautical miles; time in hours; and speed in knots. Latitude and longitude are carried within the QUICK system in the following format:

North latitude	0. (equator) to +90. (North Pole)
South latitude	0. (equator) to -90. (South Pole)
East longitude	180. to 360. (Greenwich Meridian)
West longitude	0. (Greenwich Meridian) to 180.

Latitude and longitude may be input either using the above format or using the standard degree, minute, second, and direction format. If the latter mode of input is used, the coordinates are converted to the format shown above.

The Item File

After the directory has been formed, any set of attributes can be grouped together and assigned values to define items. The attribute-value pairs are collected to describe such items as targets, weapons, defensive systems, and payloads. The value defined for each attribute of an item is validated by checking the directory.

As previously indicated, a convention which allows "global" attribute definitions may be used in preparing the data base. Within the data base, items are grouped by class and by type within class. Arranged in this manner, groups of items will tend to share common attributes, i.e., a series of items representing B-52 bomber bases may all be assigned the attribute-value pair TYPE=B-52. In this case, the user may define the

attribute using a global definition instead of explicitly listing the attribute as an item entry for each bomber base. Once defined, the global attribute remains in effect until: (1) it is removed by an UNDEFINE command inserted in the data base by the user; (2) the target class changes; or (3) the global definition is overridden by a definition contained in the incoming item. In the latter case, the global value is not applied to the item being processed, but it remains in effect for subsequent items in the class. The procedures for establishing global definitions within the data base are described in chapter 2 of the User's Manual, Volume II (see Program QUIKBASE, SETID Option, Item Cards).

QUICK Data Classes

The information included in the data base is categorized by CLASS, e.g., bombers, and by TYPE within class, e.g., B-52. Fifteen classes may be used to describe the targetable-type installations included in the data base. The data categories currently associated with each target class are shown in table 3. These classes are identified within the system by referencing the value of the attribute ICLASS, the class number. The class name, attribute CLASS, is not used for internal identification but is included in many of the printouts output by the system. The class names may be changed by the user; however, the program functions unique to a particular ICLASS must be considered in determining the class assignment for targetable items. The attributes assigned to the items of ICLASS 1 through 5 and ICLASS 14 define the item not only as a target but include the attributes which establish its offensive/defensive capabilities and characteristics.

In addition to these target classes, nine auxiliary data classes are used to enter weapon-type data such as the specific composition of a bomber payload (bombs, air-to-surface missiles (ASMs), electronic countermeasures (ECM), and decoys) and geographic-type data required by the system. Geographic-type data are required to define air defense zones and to provide for bomber routing. For example, within the Plan Generator, penetration and depenetration routing of aircraft is controlled through the use of corridors established by the planner. The data defining each of the route points associated with these corridors are included in the auxiliary class segment of the data base. Internally, these classes are identified by the class name shown in table 2 (ICLASS is not assigned to these classes). Hence, these class names may not be changed by the user.

Table 2. QUICK Classes
(Sheet 1 of 2)

TARGET CLASSES

ICLASS	CURRENT CLASS MNEMONIC	DATA CATEGORY
1	MISSILE	Offensive missiles
2	BOMBER	Offensive bombers
3	TANKER	Tankers
4	DEFCONTR	Defensive command and control
5	INTCPTOR	Interceptor aircraft
6	C/C	Offensive command and control
7	NUCSTOR	Nuclear storage sites
8	AIRFIELD	Airfields
9	NAVAL	Naval targets
10	TROOPS	Troops
11	COMMUN	Communications
12	MISC	Miscellaneous
13	U/I	Urban/industrial targets
14	ABMDEF	Area ABM defense components
15		Reserved for future use

Table 2. (cont.)
(Sheet 2 of 2)

AUXILIARY CLASSES

CLASS MNEMONIC	PURPOSE
WARHEAD	Provides warhead characteristics, e.g., yield
ASM	Provides characteristics of air-to-surface missiles, e.g., speed, warhead
PAYLOAD	Identifies weapons and penetration aids carried by a missile or bomber
DBLDATA	Contains time-dependent destruction-before-launch (DBL) probability tables
ZONE	Establishes the area of bomber air defense zones
POINT	Provides latitude and longitude for zone, route, and refuel points
BOUNDARY	Establishes the boundaries of bomber air defense zones
CORRIDOR	Identifies penetration corridors and associated attrition parameters
LEGS	Defines the penetration and depenetration route legs

OPERATIONAL RESTRICTIONS

Data Constraints (Upper Limits)

There are no restrictions on the quantity of data that may be maintained in the data library (data base DATADB). There are, however, constraints (upper limits) on the amount of data the QUICK system can process in a single run. These constraints are directly related to the storage requirements outlined above and similar requirements associated with the programs of the other QUICK subsystems. Table 3 provides a list of major constraints associated with the Data Input subsystem. A more complete list of QUICK system upper limits is contained in User's Manual, Volume II, (table 1, Major Constraints of the QUICK System).

As shown in table 3, the game data base may contain up to 12,000 items. This includes both Red and Blue items. However, only 5,000 targets per plan can be accommodated. The latter restriction is imposed by the Plan Generator and must be considered in forming the data base. For plan generation, target complexes consisting of multiple target class items are treated as a single target; experience with the NMCSSC QUICK data base indicates that at least 20% of the data base items will be assigned to target complexes (criteria explained in chapter 3). Hence, 6,000 target class items can usually be accommodated within the 5,000 target per plan limitation. Considering current QUICK support requirements, none of the existing constraints is considered a significant restriction

Table 3. Input Subsystem Major Data Limits

TARGET CLASS DATA	MAXIMUM	CODE*
Target Classes	15	1
Target Types	250	1
Target Types per Class	40/20**	2
Targets (Target Class Items)	12000	1
Targets per Plan (Red or Blue)	5000	2
Targets per Earth Sector	4000	1
Targets - Collocated	4000	1
Targets per Collocation Island	100	-
Target Complexes	4000	1
Target Elements per Complex	40	-
Targets Defended By Terminal Antiballistic Missile Interceptors	500	1
AUXILIARY CLASS DATA		
Warhead Types	50	1
Payload Types	40	2
Air-to-Surface Missile (ASM) Types	20	1
Air Defense Zones (Bomber)	63	2
Command/Control Regions	20	2
Corridors (Penetration)	30	2
Depenetration Points	50	2
Recovery Bases (Bomber) per Depenetration Point	4	-
Refuel Points (User Directed)	20	2
Route Legs	200	2
Route Points	200	2
Zone Boundary Legs	200	2

*1 = Total in game base; 2 = Total per side; - = As stated

**Missile and Bomber, 40 each; all others, 20 each

CHAPTER 3
METHODS USED

The Data Input subsystem is designed to minimize the manual effort required to prepare a QUICK game data base and the associated data files which provide data base information to the Plan Generation, Simulation, and Data Output subsystems.

As previously indicated, the Input subsystem does not involve the use of analytical techniques. Computations performed within the subsystem are not complex. Mathematical operations performed for the purpose of establishing or changing the value of selected attributes involve simple multiplication and division accomplished on the basis of a specific user-input parameter.

A detailed description of the processing and programming techniques associated with the Data Input subsystem is presented in the Programming Specifications Manual, Volume I. Specific instructions relating to the required user-input parameters, the available program options, and output prints are described in chapter 3 of the User's Manual, Volume I, and chapter 2 of the User's Manual, Volume II.

The following sections of this chapter provide an explanation of the major steps and functional requirements associated with creating, updating, modifying, and indexing a QUICK game data base.

CREATION OF GAME DATA BASE (QUIKBASE)

The creation of a game base file is performed by program QUIKBASE. It defines the data base to be used by succeeding programs of the QUICK system. This program accepts an input data base (also referred to as a data library) that specifies the attribute-value pairs for each item in the data base. This library (called DATADB) is processed to produce a game base file (called QUIKDB) which contains these attribute-value pairs in a compact format that can be used easily by the later programs.

There are two subsidiary purposes for program QUIKBASE. First, it provides a capability to create or update a data library tape. This creation or update process may be independent or may include the creation of a game base file (QUIKDB). Second, the program provides facilities to print the game base file (QUIKDB) in a format meaningful to the user.

Data Sources for Preparing the NMCSSC QUICK Data Base

The three principal data sources used in assembling the data maintained in the NMCSSC QUICK data base (DATADB) are the Joint Resource Assessment Data Base (JAD), the air order of battle files (AOB files) prepared by the Defense Intelligence Agency (DIA), and various manually prepared inputs provided by the SAGA. While the majority of the required data are retrieved from the existing automated source files maintained by NMCSSC, certain data are related directly to the plan/simulation being developed or to the QUICK techniques used in plan generation and are therefore not available from these files. For example:

1. The specific weapons assigned to each type delivery vehicle and the staging of bombers at various airfields

2. The relative strategic value of each potential target

3. The characteristics of the missile and bomber delivery vehicles, e.g., speed, range, and delivery error

4. The relative effectiveness of missile and bomber defensive forces.

Data such as that described above must be secured from available planning documents or established on the basis of user judgment and added to the basic source data. Usually, this type information is supplied by SAGA and added to the data base using the update capabilities of program QUIKBASE.

The data library tape (DATADB) is formatted for ease in updating the information contained thereon. Basically, the tape consists of a series of card images with three identifiers for each card. The identifiers are a set number, a line number, and a date. The line numbers run consecutively (increasing) within each set. The set numbers are strictly increasing through the base. Thus, to change the information on the data library file, the user specifies the set number and line number for the change and the new information to be placed at that location in the file.

Concept of Operation

There are three sources of information to program QUIKBASE: a data library file (DATADB on tape; DATAFIL on cards), user-input parameters which control the functions performed by the program, and user-input data which update the data library file. Using the parameters which control the functions, the program modifies the data library file according to the update input data, creates a game base file (QUIKDB), and prints the contents of that file.

The major options available to the user in QUIKBASE are:

1. Selection of the data library input medium

   a. Data library tape (DATADB) (default option)
   b. Punched cards } DATAFIL
   c. BCD card image tape }

2. Selection of the game base print format

   a. Consolidated print format
   b. Detailed single item prints

3. Selection of sections of updated data library (DATADB) to be printed

   a. All sets
   b. Specified sets
   c. Updated sets only

DATA BASE MODIFICATION (BASEMOD)

When the QUICK data base is created by merging data retrieved from automated data management systems, some of the data required only for the QUICK system are not present in the data base. In addition, the data base may need to be adapted to the specific scenario for which the plan is being developed. Program BASEMOD alters the content or characteristics of a data base in order to perform this adaptation.

The modification can be performed on the basic game data base produced by program QUIKBASE or the indexed data base produced by program INDEXER. The former option is used for major modifications and/or additions to the basic game data as required by the QUICK system and the desired plan scenario. The post-INDEXER mode of operation is used for minor plan variations when the planner does not wish to disturb the basic plan indexing scheme.

The major data augmentation tasks performed by program BASEMOD are:

1. Targets which are inappropriate for the plan under consideration, i.e., those targets assigned the attribute RESERVE=0, are excluded from further consideration

2. The appropriate number of weapon vehicles per bomber squadron (NOPERSQN) is chosen, depending upon the particular plan being developed (retaliatory, initiative, or surprise)

3. The number of bombers in commission (NOINCOM) for each squadron is calculated by specifying that NOINCOM is equal to a user-specified fraction of NOPERSQN

4. The number of bombers which are on alert (NOALERT) for each squadron is calculated by specifying that NOALERT is equal to a user-specified fraction of NOINCOM

5. The appropriate relative effectiveness (EFECTNES) of each interceptor squadron and each defensive command/control installation is selected based on the type plan being developed

6. The relative value (VAL) of urban/industrial targets is calculated as a function of general industrial worth (IGIW) and population (POP) according to the formula VAL = A·IGIW + B·POP, where A and B are player inputs

7. If required, target defenses (TARDEFs) are processed

8. If required, air defense zones are calculated for classes 4 and 5. These zones are usually already part of the data base prepared for QUICK.

For input, the Plan Generation subsystem uses subsets of the data that are contained in the general data base. The exact composition of these subsets depends upon which type of plan is being developed. Thus, there is a need to examine in detail all of the attributes contained in the basic game data base and, depending upon which of the three possible plans is to be constructed (retaliatory, initiative, or surprise), to omit from further consideration those data which are not relevant.

This task is accomplished by running program BASEMOD after program QUIKBASE. The program sequentially examines each item in the data base file. According to user-specified input parameters, items with the attribute RESERVE equal to zero are omitted; selection is made of the correct number per squadron, number in commission, number on alert, and effectiveness; and adjustment is made of the attribute VAL as a linear combination of IGIW and POP.

When run after program INDEXER, program BASEMOD examines items in the indexed data base and deletes items on the basis of their country location code, CNTRYLOC. The user can thus provide for variations in the target system by country without disturbing the basic indexing scheme.

SUMMARIZING THE GAME DATA BASE (BASESUM)

Program BASESUM summarizes game data bases and prints these summaries in tabu ar form. Program BASESUM may be used to summarize the data base contained on the output tapes produced by programs QUIKBASE, BASEMOD, or INDEXER. The only input required by the program is a data base tape. The output consists of printed summary information. There are no user-input parameters required for this program.

Program BASESUM makes two passes through the data base and summarizes the contents by side and class. For each side-class combination, one table is generated in which the columns are the types found and the rows are all the attributes defined for this class. One line called ITEMS is added to the row heading, although it is not a true data base attribute. It contains the count of the number of items of that side, class, and type.

SUBSYSTEM INTERFACES WITH DATA INPUT SUBSYSTEM (INDEXER)

In order to provide for efficient data handling and communication between the programs of the QUICK system, it is necessary to assign indices to the data in the data base. Program INDEXER of the Data Input subsystem is designed to perform this important function.

The input to program INDEXER consists of either the game data base as created in program QUIKBASE (QUIKDB) or the modified data base (QKMODDB) as constructed in program BASEMOD. The output from the program consists of an indexed data base tape, INDEXDB, which provides data used by the Plan Generation and Output subsystems, and a simulation data tape, SIMTAPE, which is used in program SIMULATE of the Simulation subsystem.

The indexed data base contained on INDEXDB is normally input directly to the Plan Generator. However, if required, additional modifications to this file may be implemented using program BASEMOD. If this option is exercised, INDEXDB becomes an input to program BASEMOD, which prepares a modified indexed data base INMODDB for input to the Plan Generator.

The data base on the INDEXDB tape differs from the data base which is input to INDEXER (QUIKDB or QKMODDB tape) in the following ways: the indices INDEXNO, ITYPE, and JTYPE are defined for all appropriate items, and a positive value of the attribute ICOMPLEX is assigned to all elements of the target complexes formed by INDEXER.

The SIMTAPE is a library type file. It provides information which supplements the war plan data provided by the Plan Generator and is required by the Simulator to model and evaluate the results and interactions of the planned events. The SIMTAPE information includes:

1. The array COLAR which provides information on collocated targets, e.g., the index number INDEXNO and the relative coordinates (offset distances in units of .02 nautical miles) between collocated targets.

2. The array STATUS which provides information on every target class item processed by INDEXER. This array consists of one word for each value of INDEXNO. A series of codes and indices packed within each word provides: (1) information on the target (INDEXNO item), e.g., its collocation status (one if collocated, otherwise zero); and (2) indices to other arrays containing target information, e.g., the index to the array VULN which contains the target's vulnerability number.

3. Several tables which provide (1) type characteristics, i.e., data such as probability of a launch abort, which are the same for every vehicle of a given type and (2) data establishing the defensive capabilities of each side, e.g., the defensive potential of each bomber air defense zone.

Indexing Operations

To facilitate cross-referencing between data base items, all items in the target classes are assigned indices. Each item is assigned a unique index number (attribute INDEXNO) and two indices, ITYPE and JTYPE, which identify the item as a member of a specific "type set."

Program INDEXER assigns indices to the various kinds of data as follows. For all classes that contain items which could be considered to be targets, the user assigns a value of the attribute ICLASS from 1 to 15 at the time the data base is prepared. All target types (attribute TYPE) which belong to these classes are assigned distinct values of the attribute ITYPE, and all types within each class are assigned distinct values of the attribute JTYPE. Each item in classes 1 through 15 is then assigned a distinct value of the attribute INDEXNO; this assignment preserves the partial ordering of the items due to the ITYPE assignment.

The assignment of these indices is performed by program INDEXER which makes several passes through the game data base file. In pass one, the values of ITYPE, JTYPE are assigned, and other indexing operations are completed as follows.

The correct number of individual sites for each missile squadron are grouped together on the input data base tape, and the attribute ISITE is set to 1 for the first site appearing on the tape in each squadron. Program INDEXER then assigns consecutive values of ISITE to the remaining sites in each squadron. Missile sites (ICLASS=1) for which ISITE equals 1 and all items in classes 2 through 15 are assigned temporary values of JTYPE and are counted in the following manner. The array TYPENAME(J,ICLASS) is searched for a match with the attribute TYPE of the current item. If no match is found, then the first blank word in the array TYPENAME(J,ICLASS) is filled with TYPE ($1 \leq J \leq 40$ for SIDE=BLUE, $41 \leq J \leq 80$ for SIDE=RED). The attribute JTYPE is given the value J, and the counter TYPETBL(J,ICLASS) is increased by NADD, where NADD is the number of sites per squadron if CLASS=MISSILE, and NADD=1 otherwise.

When all items have been processed, the array TYPENAME is collapsed so that the types are listed consecutively as established by the (JTYPE,ICLASS) assignment. The new indices of the types in TYPENAME will be assigned to all appropriate items as values of ITYPE during the next pass through the data base. The items are written on a temporary data base file.

In pass two, the data base which was written during the first pass is read, and those items for which ICLASS is not defined (auxiliary classes) are copied onto another temporary data base file. Those items in classes 1 through 15 are first assigned values of ITYPE (as described above), then assigned values of INDEXNO in the following manner. For the first item of a given type, INDEXNO is defined to be INDBEG(ITYPE); then succeeding items of this type are given progressively larger values of INDEXNO.

To provide convenience in handling arrays which will be indexed by JTYPE, items with SIDE=RED are reassigned values of JTYPE contiguous with the values for BLUE items. For example, if there are a total of 15 target types on the BLUE side and 10 types on side RED, the BLUE types are numbered 1 to 15 consecutively and the RED types are numbered 16 to 25. The first RED type is 16, the second is 17, and so on.

Index Breakpoint Tables

Every item in the game data base which is given a value of ICLASS is also assigned a unique value of the attribute INDEXNO, which is a unique identifier for that item for game purposes. The order of indexing is as follows: first, all items of the same class are numbered consecutively. Within a single class, items are grouped according to the attribute SIDE (value RED or BLUE). Items are further grouped according to type. Within a type, items are assigned index numbers according to the order in which they appear in the data base.

To facilitate cross-referencing in subsequent programs, program INDEXER constructs a set of breakpoint tables which reflect the indexed structure of the data base. With the aid of these breakpoint tables, which give the beginning indices INDEXNO of each class and type and the number of RED and BLUE types in each class, it is possible to obtain the class, type, and side of any item from its index number. The breakpoint tables are included on the output tapes prepared by program INDEXER (INDEXDB and SIMTAPE). A more complete description of these tables is provided in the User's Manual; Volume I (see chapter 3, Data Input Subsystem, Program INDEXER, Output Print Option 1).

Collocation Islands and Complex Targets

In order to allow more efficient operation of the Plan Generation and Simulation subsystems, the individual target items in the game data base are aggregated into "clusters." These clusters may take two forms, collocation islands or complex targets.

Collocation Islands: To facilitate the calculation of target kill probabilities during simulation, the QUICK design provides for grouping appropriate targets into collocation islands. Collocation islands are defined by the following criterion: if the distance between two targets is less than the sum of the lethal radii of a one-megaton weapon (an arbitrary selection), considering the hardness of each target, the targets belong to the same collocation island. A collocation island consists of all targets (up to 100) which are linked by this distance criterion. In practice, islands are usually rather small clusters. Targets are said to be collocated if they belong to the same collocation island; a collocated target is one which belongs to some collocation island. During simulation, if a weapon is successfully delivered against a collocated target, the effects of the burst are assessed against all targets in the collocation island. Data reflecting the relevant target-to-target spacing for each collocation island are referenced by the Simulator through the use of the index number INDEXNO assigned to each target.

Complex Targets: The definition of a complex target is identical to the definition of a collocation island, except that the required distance between targets is one-half the distance defined for collocation. Therefore, every complex target is a subset of some collocation island. Thus, complex targets consist of target elements (up to 40 data base items) which are either exactly collocated or within the defined distance. Under this criterion, the target elements are reasonably close together and should be considered as a unit. Therefore, the Plan Generator deals with target complexes, i.e., a complex target, as a single simple target during the weapon allocation phase (see Analytical Concepts and Techniques. Target List Preparation, in chapter 2, Analytical Manual, Volume II).

Processing Procedure: The formation of these target clusters is performed by program INDEXER. The INDEXNO, LAT, LONG, and a distance are stored for each item for which INDEXNO is defined in the data base. These data are used in making up collocation islands and complex targets. The distance is the lethal radius of a one-megaton weapon corresponding to the vulnerability for the target. The restriction on available storage in the computer memory requires that most of the collocation data must be temporarily stored on the disk.

After all items in the data base have been processed, an INDEXER subroutine (COLOCATE) is called, and collocation islands and complex targets are constructed from the items for which data are currently held in memory. When control is returned from the subroutine, the information for the next sector is read, and the subroutine is called again. This process is repeated until all sectors have been investigated. The operation of the collocation process is described below.

Calculations: Program INDEXER makes up collocation islands and complex targets for a list of as many as 4,000 targets in any single earth sector. For each target, INDEXNO, LAT, LONG, and the critical distance for collocation which was described earlier are stored in the arrays IND, Y, X, and Z, respectively. The array Z actually contains the distances in degrees of latitude, so that computation is minimized during execution.

The description of collocated and complex targets is contained in a series of arrays which are used to cross-reference the target list.

The array COMPLEX and the logical arrays COLO and COMP are used to record the status of each item: COLO and COMP are indexed by INDEXNO and are set to 1 to indicate that the corresponding item belongs to a collocation island or a complex target, respectively; COMPLEX is a packed array containing INDEXNO and ICOMPLEX for each element of a complex target. The logical arrays COL, CL, CLT, CP, and the array LCOMP are used within COLOCATE to maintain the status of the items currently being investigated. The indices to the four logical arrays correspond to the indices for IND; the value 1 for COL(J), CL(J), CLT(J), or CP(J) indicates that the Jth item in the array IND is a member of some collocation island, a member of the island currently being investigated, a member of the current island which has not yet been checked for further collocation, or a member of a complex. The appearance of a number J in the array LCOMP indicates that the Jth item in IND is a member of the complex target currently being investigated.

The initial operation of the collocation process is to arrange the arrays IND, Y, X, and Z so that X (longitude) is ordered by increasing magnitude, and IND, Y, and Z are in the corresponding order.

The search for collocation begins by comparing the first element in the list with the following elements; if the difference in longitude is sufficiently small, the actual distance is calculated and compared with the sum of the critical distances for the two targets. When two targets are found to be collocated, COL and CL are set to 1 for both, and CLT is set to 1 for the second. If they are sufficiently close to be elements of a complex target (one-half the distance for collocation), CP is set to 1 for both, and both indices are entered in LCOMP. When a difference in longitude is encountered which is too great, processing of the item being investigated is considered finished, and CLT is set to 0 for that item. If the item is a member of a complex target, the next member of that complex in the list LCOMP is compared in the same way with all other items to find additional members of the complex; this process is repeated until all items in LCOMP have been investigated, and the complex is finished. The complex is assigned a value of ICOMPLEX which is packed along with IND(J), i.e., INDEXNO, into consecutive words in the array COMPLEX. Next, LCOMP is cleared and investigation of the current collocation island is continued, beginning with the next item for which CLT is 1. When the collocation island is finished, the required data for the items which belong to the island are packed in the array COLAR, CL is reset to 0, and COLAR is written on disk. Then the next item in the list for which COL is 0 is compared with all others to restart the investigation. When the list is exhausted, all collocation islands and complex targets for the current earth sector have been processed. Subsequent earth sectors are processed in the same manner until all data base target items have been checked.

During the last pass through the data base, as each item is processed, those which are elements of complex targets (as indicated by the flag COMP described above) are assigned the value of ICOMPLEX and written on the INDEXDB tape. The complex target data are used by the Plan Generator (see Analytical Concepts and Techniques, Target List Preparation, in chapter 2, Analytical Manual, Volume II). The collocation data contained in the COLAR array are subsequently written on the SIMTAPE for use during the simulation phase.

Preparation of Simulation Data

The data base information required by the Simulator is transmitted via the SIMTAPE prepared by program INDEXER. The SIMTAPE contains only that data required for simulation (a significant segment of the data included in the data base is required by the Plan Generator but not used by the Simulator). These data are organized as a series of indexed tables (arrays) to enhance storage and retrieval during the simulation phase. On option, the contents of each SIMTAPE table may be printed (see Program INDEXER, Output Print Options, in chapter 3 of the User's Manual, Volume I).

A detailed description of content and format of the SIMTAPE is presented in the Programming Specifications Manual, Volume I (Chapter 7, Program INDEXER, Output Files).

CHAPTER 4
ACCURACY

The programs in the Data Input subsystem perform mainly data processing functions wherein computational accuracy is not a likely source of error. None of the calculations which are performed in this subsystem are complex. The CDC 3800 computer word structure is large enough to contain all of the significant digits necessary for proper execution of the system. The only problems which may affect the operations of these components with respect to the remaining programs in the system are input and output functions. These are discussed below.

Input

Inputs to program QUIKBASE of the Data Input subsystem do present an operational problem with respect to accuracy. The program now accepts data in the first eight columns of each 10-column field in a card image. This means that all attributes and their assigned values must be represented in eight characters or less.

Since many of the quantities (values of attributes) are included in data tapes prepared for subsequent processing, the level of significance on the accuracy results from expressions using these attributes will be correspondingly limited. This is not believed to represent a significant limitation on the capabilities of the QUICK system.

Output

The data output from the Data Input subsystem and the various programs internal to the subsystem will be as accurate as that data input to program QUIKBASE.

The Data Input subsystem provides listings of data which have been processed by the subsystem. These listings reflect the stored values of attributes at the time they are printed. In some cases, when a real number (floating point) exceeds the expected output range (the format indicated in the directory portion of the Data Base File) the program in which the print is requested indicates that condition by placing all asterisks in the output field. Thus, the presence of asterisks in the output indicates incompatibilities between the data being printed and the format used in printing the data. This in no way reflects a deviation of accuracy in the data quantities. If an integer (fixed point) number

exceeds the expected output range, the CDC 3800 SCOPE operating system software removes the most significant digits (including sign) from the output field with no indication. For example, if the number 12345 were to be printed under FORTRAN format I2, the digits 45 would appear in the output listing. In many cases the format in the directory, for example, can accommodate only eight characters, a limitation on the input procedures of the Data Input subsystem. This limits the number of characters which can be printed but does not limit the quantity in memory or on the data files being processed or generated.

APPENDIX A
GLOSSARY

Alphabetic-numeric: The characters which include letters of the alphabet, numerals, and other symbols such as punctuation or mathematical symbols.

Alphameric: A contraction of alphabetic-numeric.

Alphanumeric: A contraction of alphabetic-numeric.

Array: A series of items arranged in a meaningful pattern.

Bit: An abbreviation of binary digit. A single character in a binary number.

Circular Error Probable (CEP): An indicator of the delivery accuracy of a weapon system, used as a factor in determining probable damage to a target. It is the radius of a circle within which half of the missiles/ projectiles are expected to fall.

Computer Program: A program expressed in computer code designed to solve a class of problems, or specializing on a specific problem when appropriate parametric values are supplied.

Computerized War Gaming Model: A computer program, or series of programs, designed to simulate the logic of actions or interactions of a conflict situation and provide results for subsequent analysis.

Damage Expectancy (DE): Probability of achieving a desired level of damage considering the probability of weapon arrival (PA) and the probability of damage (PD), i.e., DE=PAxPD.

Data Base: An organized collection of data records with similar or associated characteristics either to be operated upon by a system or contributing to the operation of a system.

Data Library: A collection of information available to a computer.

ESP Model: The Event Sequenced Program used by the Joint Strategic Target Planning Staff (JSTPS) to simulate large-scale strategic warfare.

Event: A happening in time, either within a simulation or in reality.

Expected Value: The average or mean value which would be obtained if a given event were repeated many times.

FLAG: A code used in imposing restrictions on the allocation of weapons within QUICK.

General War: Armed conflict between major powers in which the total resources of the belligerents are employed, and the national survival of a major belligerent is in jeopardy.

Input: Any factors, data, parameters, values, or instructions required for proper operations of a model or submodel to produce game results.

JAD: Joint Resource Assessment Data Base. The JAD is an automated repository of information acquired from several sources which is stored and maintained by NMCSSC.

Library Tape, QUICK System: A magnetic tape on which the programs and data handling routines of the QUICK system will reside.

Magnetic Tape: A computer storage device in which data are stored in the form of magnetic spots on metal or coated plastic tape.

NEMO Model: The Nuclear Exchange MOdel maintained by the Navy for simulation of a two-sided global nuclear war.

Nuclear Vulnerability Assessment: The estimation of probable or expected effects of hypothetical nuclear attacks on population, forces, and resources.

Posture: Relative place or position; state or condition at a given time, especially in relation to other persons or things.

Probability of Damage (PD): The probability that damage will occur to a target expressed as a percentage or as a decimal.

REST-III Model: REsource STatus Damage Assessment Model III.

RISOP: A hypothetical Red Integrated Strategic Offensive Plan.

SIDAC: Single Integrated Damage Analysis Capability System.

SIOP: The Single Integrated Operational Plan.

TASK: A two-character descriptive code assigned to all targets.

APPENDIX B
QUICK ATTRIBUTE NAMES AND DESCRIPTIONS

ATTRIBUTE NAME	DESCRIPTION
ABRATE	Probability of aircraft in-flight abort per hour of flying time
ADBLI	ALERTDBL probability for initiative attack
ADBLR	ALERTDBL probability for a retaliatory attack
ADEFCMP	Area ballistic missile defense (BMD) component index (radar or missile launch site)
ADEFZON	Area ballistic missile defense (BMD) zone number
AGX	Offset X-coordinate of AGZ (fiftieths of nautical miles)
AGY	Offset Y-coordinate of AGZ (fiftieths of nautical miles)
AHOB	Actual height of burst of weapon (air or ground)
ALERTDBL	Probability of destruction before launch (DBL) of alert delivery vehicle (missile or bomber)
ALERTDLY	Delay of alert vehicle before commencing launch (hours)
AREA	Area of a bomber defense ZONE (millions of nautical miles^2)
ASMTYPE	Air-to-surface missile type
ATTRCORR	Attrition parameter for a bomber corridor (probability of attrition per nautical mile)
ATTRLEG	Attrition parameter for each route leg in bomber sortie (probability of attrition per nautical mile)
ATTRSUPF	Amount of original attrition that remains after defense suppression

ATTRIBUTE NAME	DESCRIPTION
AZON1	First area defense zone covered by a BMD long-range radar
AZON2	Second area defense zone covered by a BMD long-range radar
AZON3	Third area defense zone covered by a BMD long-range radar
BCODE	Code indicating the outcome of a simulated bomber event
BENO	Bombing encyclopedia number
BLEGNO	Index to boundary line segment
CATCODE	Category Code as reflected in Joint Resource Assessment Data Base (JAD)
CCREL	Regional reliability of offensive command and control (probability)
CEP	Circular error probable (CEP), delivery error applicable to bomber and missile weapons (nautical miles)
CLASS	Class name assigned identify sets of TYPES in data base
CLASST	Target CLASS
CNTRYLOC	Country code for country where item is located
CNTRYOWN	Country code for country which owns the item
CNTYLOCT	Target country code for country where the target is located
CNTYOWNT	Target country code for country which owns the target
CODE	Outcome code for a general event used in simulation
CPACTY	Capacity of a bomber recovery base (number of vehicles)

ATTRIBUTE NAME	DESCRIPTION
DATEIN	Earliest date in inventory (year)
DATEOUT	Latest date in inventory (year)
DEFRANGE	Typical range of interceptors at defense bases near a corridor (nautical miles)
DELAY	Delay time (e.g., launch delay time) (hours)
DELTA	Time interval between successive vehicle launches from the same base (missile or bomber) (hours)
DESIG	Target designator code, e.g., AB100, which uniquely identifies each target element included in the data base
DGX	Offset X-coordinate of desired ground zero (DGZ) (fiftieths of nautical miles)
DGY	Offset Y-coordinate of DGZ (fiftieths of nautical miles)
DHOB	Height of burst of weapon (0-ground, 1-air)
EFECNES1 EFECNES2	Attributes assigned to command and control installations and fighter interceptor units: the value EFECNES1 or EFECNES2 assigned to attribute EFECTNES depending on value of BASEMOD input parameter POSTURE (if POSTURE=1, EFECNES1 is used; otherwise EFECNES2 value is assigned)
EFECTNES	Air defense capability (arbitrary scale) established by user to indicate relative effectiveness of air defense command and control installations and fighter interceptor bases
EVENT	Index to event type
EVENTN	Index to type of event which did not occur
FFRAC	Fission fraction (fission yield/total yield)
FLAG	Numeric code (1 through 9 permitted) used to impose restrictions on the allocation of weapons within QUICK

ATTRIBUTE NAME	DESCRIPTION
FLTNO	Flight number for a sortie
FUNCTION	Operational application code for a weapon system (e.g., ICBM)
FVALH1	Fraction of value of target in first hardness component
FVALT1	Fraction of target value that disappears by T1 (percent)
FVALT2	Fraction of target value that disappears by T2 (percent)
H1	First hardness component of a target (VULN)
H2	Second hardness component of a target (VULN)
HILOATTR	The ratio of the low-altitude attrition rate to the high-altitude rate (decimal fraction)
IALERT	Alert status; 1 = alert, 2 = nonalert
IALT	Altitude index (1 = high, 0 = low)
IATTACK	Selection index for preferential area BMD; 1 forces target selection for defense.
ICLASS	Class index assigned for game
ICLASST	Target class index
ICOMPLEX	Complex index
ICORR	Bomber corridor index number assigned in program PLANSET: 1 - Tactical (FUNCTION=TAC) aircraft corridor (TYPE name DUMMY in the data base) 2 - Naval attack corridor (TYPE name NAVALAIR in the data base) used by bomber units with PKNAV greater than zero >2 - Other corridors used by long range bombers (FUNCTION=LRA)

ATTRIBUTE NAME	DESCRIPTION
IDBL	Index to data tables for time-dependent destruction before launch probability
IDUD	Dud warhead indicator; assigned to weapons which arrive at the target but fail to detonate; 1=dud warhead
IGIW	Indices of General Industrial Worth (IGIW) (dollars)
IGROUP	Group index assigned for weapon grouping during game
IMIRV	Identifying index for system with multiple independently targetable re-entry vehicles
INDEXNO	Index of a data base item (potential target) used during processing to identify the item
INDV	Vehicle index within base
INTAR	Target index (corresponds to INDEXNO)
IPENMODE	Penetration mode; 1 = aircraft uses penetration corridor, 0 = penetration corridor not used
IPOINT	Index to a geographic point
IRECMODE	Recovery mode; 1 = aircraft should plan recovery, 0 = aircraft recovery not planned
IREFUEL	Bomber refueling code
IREG	Index to identify a geographic region
IREP	Reprogramming index (capability of missile squadron)
ISITE	Site number
ITGT	Target index number assigned by Plan Generation subsystem
ITIME	Index to time periods in time dependent DBL data tables

ATTRIBUTE NAME	DESCRIPTION
ITYPE	Type index assigned for game
ITYPET	Target type index
IVULN	Index to vulnerability number table
IWTYP2	Second warhead type
JTYPE	Type index within class
JTYPET	Target type index within class
KORSTYLE	Parameter to adjust mode of corridor penetration
LAT	Latitude (degrees)*
LEGNO	Index to line segment
LINK	The index of a leg linked to the current point
LONG	Longitude (degrees)*
MAJOR	Major reference number as reflected in the Joint Resource Assessment Data Base (JAD)
MAXFRACV	Maximum value of weapon resources to be used relative to target value (in processing MAXCOST=MAXFRACV)
MAXKILL	Desired maximum damage expected for a target
MINKILL	The required minimum damage established for a target

---

* Latitude and longitude are carried internally in the QUICK system in the following format:

North latitude	0. (equator) to +90. (North Pole)
South latitude	0. (equator) to -90. (South Pole)
East longitude	180. to 360. (Greenwich Meridian)
West longitude	0. (Greenwich Meridian) to 180.

These attributes may be input in either the above format or in standard degree, minute, second, direction format.

ATTRIBUTE NAME	DESCRIPTION
MINOR	Minor reference number as reflected in JAD to identify an item
MISDEF	Number of terminal ballistic missile interceptors for a target
MVA	Manufacturing value added (MVA); indicates the amount of value added by manufacture within a specific area (expressed in U.S. dollars)
MWHDS	Number of missile warheads penetrating area defenses to terminal defense
NADBLI	NALRTDBL for initiative attack
NADBLR	NALRTDBL for retaliatory attack
NAINT	Number of area ballistic missile interceptors at an interceptor launch base
NALRTDBL	Probability of destruction before launch (DBL) of non-alert vehicle
NALRTDLY	Delay of non-alert vehicle before commencing launch (hours)
NAME	Arbitrary alphameric descriptor for any item included in the data base
NAREADEC	Number of decoys per independent re-entry vehicle for area BMD
NASMS	Number of ASMs carried by a bomber
NCM	Number of countermeasures carried by vehicle
NDECOYS	Number of decoys on a bomber or number of decoys per independent re-entry vehicle for terminal BMD
NDET	Number of warheads detonating in current event
NEXTZONE	The adjacent zone to a side of a defense zone
NMPSITE	Number of missiles per site

ATTRIBUTE NAME	DESCRIPTION
NOALERT	Number of vehicles on alert at a base
NOBOMB1	Number of first bomb type carried by vehicle
NOBOMB2	Number of second bomb type carried by vehicle
NOINCOM	Number of delivery vehicles in commission
NOPERSQN	Number of weapon vehicles per squadron
NOPERSQ1 NOPERSQ2 NOPERSQ3	Attributes used in program BASEMOD to compute the value of the attribute NOPERSQN for bomber units; numbers 1, 2, and 3 specify surprise, initiative, and retaliatory attack plans respectively
NPEN	Number of warheads penetrating in current event
NTARG	Number of targets in missile launch event
NTINT	Number of terminal BMD interceptors at target
NWHDS	Number of warheads per independent re-entry vehicle (missiles)
NWPNS	Number of weapons in a group
NWTYPE	Warhead type
PARRIVE	Probability of bomber arrival in current event
PAYLOAD	Index which identifies entire weapon and penetration aid complement on a vehicle
PDES	Probability that launch failure destroys missile
PDUD	Probability a warhead will fail to detonate
PEN	Penetration probability for a weapon
PFPF	Probability of failure during powered flight (missiles)
PINC	Probability that a missile is in commission

ATTRIBUTE NAME	DESCRIPTION
PKMIS	Probability a missile fails to penetrate terminal defense
PKNAV	Single shot kill probability of a weapon against a naval target (a value greater than zero restricts weapon use to naval targets)
PLABT	Probability of vehicle launch abort
PLACE	Index to geographic location of an event
PLACEN	Index to geographic location of an event which did not occur
POP	Population (cities) (thousands)
POSTURE	Force readiness condition
PRABT	Probability of refueling abort
PRIMETAR	Prime target flag; 1 signifies priority target in a complex
PSASW	Destruction before launch probability assigned a weapon for a specified time period
RADIUS	Size descriptor for area targets (nautical miles)
RANGE	Vehicle range (nautical miles)
RANGEDEC	Range decrement for low-altitude aircraft flight (high range/low range)
RANGEREF	Range (nautical miles) of bomber with refueling
REL	Reliability - probability that weapon system will arrive at target given successful launch
RESERVE	Technique used to remove certain targets from weapon allocation when RESERVE = 0
SIDE	Item side name, currently either "RED " or "BLUE"

ATTRIBUTE NAME	DESCRIPTION
SITENO	Site number (currently for individual missile sites)
SPDLO	Speed at low altitude (knots)
SPEED	Speed (knots)
SQNNO	Squadron number
T1	Time of departure of first value component of a target
T2	Time of departure of second value component of a target
T3	Time of departure of third value component of a target
TAIM	Number of aim points perceived by terminal defense in current event
TARDEFHI	Level of local bomber defense at high altitude*
TARDEFLO	Level of local bomber defense at low atltitude*
TASK	Target task code indicating targeting priority
TGTSTAT	Indicates target status as dynamic or nondynamic; in simulation status (alive/dead) is maintained for dynamic targets
TIME	Game time at which event occurred (hours)
TIMEN	Time planned for event which did not occur (hours)
TMDEL	Mean delay time to relaunch after a nondestructive aircraft abort (hours)

* Arbitrary units scaled by user-input parameter in Plan Generation subsystem. Minimum value 0 for no defense. Highest allowed defense level is + 7.

ATTRIBUTE NAME	DESCRIPTION
TPASW	Time at which a time period ends for DBL data tables; there may be up to 10 time periods for each table
TRETARG	Time required to retarget for known in-flight missile aborts (hours)
TTOS	Total time on station (for a tanker) (hours)
TVUL	Time a missile remains within vulnerable range of launch .ite (hours)
TYPE	Arbitrary alphameric designator (type name) to identify smallest sets in data base
TYPET	Target TYPE
TYPE1 TYPE2	Attributes assigned to command and control installations and fighter interceptor units: attribute TYPE assigned TYPE1 or TYPE2 value based on BASEMOD input parameter POSTURE (POSTURE=1 TYPE1 is used; otherwise TYPE2 value used)
VAL	Relative value of an item within its CLASS as established in the data base by the user
VALU	Game value of an item (assigned in plan generation based on user-input parameters)
VAL1 VAL2	Attributes assigned to command and control installations and fighter interceptor units: attribute VAL assigned VAL1 or VAL2 value based on BASEMOD input parameter POSTURE (POSTURE=1, VAL1 is used; otherwise VAL2 value is assigned)
VULN	Vulnerability number
WACNO	World aeronautical chart number
WHDTYPE	Warhead type index assigned in the data base
WHDTYPEN	Warhead type index (used with EVENTN)
YIELD	Yield (MT)
ZONE	An area bomber defense zone enclosed by a set of linked boundary points